IMPLICIT-EXPLICIT SWITCHING FOR

FINITE ELEMENT ANALYSIS

Bradley N. Maker

5

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to computer simulations using the finite element method, and more specifically, to a method for switching between the implicit

10 finite element method and the explicit finite element method during a simulation.

Related Art

The finite element method is a technique for obtaining approximate numerical solutions to boundary value problems which predict the response of physical systems subjected to external loads. The finite element method is described in detail by Thomas

15 J. R. Hughes in "The Finite Element Method" (1987), published by Prentice-Hall, Inc., New Jersey, which is incorporated herein by reference in its entirety. One common use of the finite element method is in the field of solid mechanics where it is used to analyze structural problems such as the formation of stamped sheet metal parts or the springback of stamped sheet metal parts.

20 In general, finite element analysis begins by generating a finite element model of a system. In this model, a subject structure is reduced into a number of node points which are connected together to form finite elements. Governing equations of motion are written in a discrete form, where the motions of each node point are the unknown part of the solution. A simulated load or other influence is applied to the system and

25 the resulting effect is analyzed using well known mathematical methods.

1

Two such mathematical methods are the implicit finite element method ("the implicit method") and the explicit finite element method ("the explicit method"). These methods are used to solve transient dynamic equations of motion and thus obtain static equilibrium solutions to the equations. When obtaining the static equilibrium

5   solutions, the effects of inertia are not included. Since the effects of inertia are not included, the implicit method can be used to solve the governing equations directly, but the explicit method can provide only approximate solutions. Conventional computer simulations are performed using either implicit method finite element software or explicit method finite element software.

10   The implicit method is characterized by the formation of a stiffness matrix to represent the interaction of nodal motions within the structure. In the implicit solution process, the stiffness matrix is assembled, its inverse is computed, and this inverse is applied to an array of nodal forces to produce a solution of nodal displacements. For nonlinear problems, these displacements are tested to verify that they satisfy the

15   governing equations of motion. If the equations are not satisfied, an iterative procedure is applied to refine the accuracy of the solution. When a satisfactory solution is finally obtained, the iterative process is said to have converged. Successful convergence of these iterations is not guaranteed and can be very difficult in practice.

The implicit solution procedure can be described mathematically as follows: A

20   statement of static equilibrium may be written as a balance of internal and external forces at each node point in the model

$$\mathbf{f}^{int}(\mathbf{u}) = \mathbf{f}^{ext}$$

where $\mathbf{f}^{int}(\mathbf{u})$ and $\mathbf{f}^{ext}$ are the discrete vectors of internal and external forces, and for brevity only $\mathbf{f}^{int}$ is expressed as a function of the nodal displacement vector $\mathbf{u}$.

2

At each load step (time step) during the simulation, a displacement vector is sought which satisfies equilibrium according to the above equation. The equations describing this iterative search are obtained by performing a first order Taylor series expansion about the displacement solution at iteration (i+1), and moving known terms from

5   iteration (i) to the right

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \Delta\mathbf{u}$$

$$\mathbf{f}^{int}\left(\mathbf{u}^{(i)} + \Delta\mathbf{u}\right) = \mathbf{f}^{ext}$$

$$\frac{\partial \mathbf{f}^{int}}{\partial \mathbf{u}}\left(\mathbf{u}^{(i)}\right)\Delta\mathbf{u} = \mathbf{f}^{ext} - \mathbf{f}^{int}\left(\mathbf{u}^{(i)}\right)$$

$$\mathbf{K}\left(\mathbf{u}^{(i)}\right)\Delta\mathbf{u} = \mathbf{R}^{(i)}$$

where $\mathbf{K}$ represents the stiffness matrix of the structure, $\mathbf{R}$ represents the out-of-balance or residual force vector, and $\Delta\mathbf{u}$ represents the increment or change in displacement which advances the solution from iteration (i) to iteration (i+1).

10   The solution is initialized for time step (n+1) using the equilibrium displacement solution from the previous time step (n)

$$\mathbf{u}^{(0)}_{(n+1)} = \mathbf{u}_{(n)}$$

The solution is then advanced iteratively by successively solving for the displacement increment, and then updating the total displacement

$$\Delta\mathbf{u} = \mathbf{K}^{-1}\left(\mathbf{u}^{(i)}\right)\mathbf{R}^{(i)}$$

15

$$\mathbf{u}^{(i+1)} = \mathbf{u}^{(i)} + \Delta\mathbf{u}$$

When the correct equilibrium solution is found, both the residual force vector and the displacement increment become zero. Norms of these vectors are used to evaluate the accuracy of a solution during the iteration process, and to test for

3

convergence of the equilibrium iterations. The product of displacement increment and residual force can be computed to give the incremental internal energy of the model, which can be monitored for excessive growth to detect divergence of the equilibrium iteration process.

5   An advantage of the implicit method is that a solution which satisfies static equilibrium can be computed directly. Another advantage of the implicit method is that there is no algorithmic limit to time step size. Loads can typically be applied in tens or hundreds of steps, rather than millions of steps.

        A disadvantage of the implicit method is that a stiffness matrix must be formed

10   and inverted. Assembling and inverting the stiffness matrix in a typical application requires a relatively large amount of computer CPU and memory resources. For example, in a typical automotive application, the dimensions of the stiffness matrix can be more than one million by one million. Another disadvantage of the implicit method is that convergence of the nonlinear equilibrium iteration process is not guaranteed.

15   Successful convergence at each step of the solution requires proper choice of load step (time step), and good performance of the numerical models for material behavior and contact interaction between parts. Each simulation presents a new set of difficulties, so that reliable solutions are not always available even for expert users of the method.

        The explicit method is based on a statement of dynamic equilibrium: force is

20   equal to the product of mass and acceleration. This condition is applied independently at each node point. Thus, at each node point, forces are assembled, and the acceleration of each node is computed simply by dividing by nodal mass. This process is repeated as the solution is incrementally advanced in time, integrating the nodal accelerations to obtain velocity and displacement. In order for this solution to remain accurate, a limit

25   is placed on the maximum size of the time step. This limit, known as the Courant

4

condition, is typically less than one microsecond in automotive applications. This means that over one million time steps are often required to simulate one second of structural response.

The explicit solution procedure can be described mathematically as follows: A

5   statement of dynamic equilibrium may be written as a balance between inertial, internal and external forces at each node point in the model

$$\mathbf{Ma} + \mathbf{f}^{int}(\mathbf{u}) = \mathbf{f}^{ext}$$

where $\mathbf{M}$ is the nodal mass matrix, $\mathbf{a}$ is the nodal acceleration vector, and

$\mathbf{f}^{int}(\mathbf{u})$ and $\mathbf{f}^{ext}$ are the discrete vectors of internal and external forces, and for brevity

10   only $\mathbf{f}^{int}$ is expressed as a function of the nodal displacement vector $\mathbf{u}$. The solution is advanced by solving the above equation for the acceleration at time (n), and then applying the central difference method to obtain nodal velocity and displacement vectors

$$\mathbf{v}^{(n+1/2)} = \mathbf{v}^{(n-1/2)} + \mathbf{a}^{(n)}\Delta t$$

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n-1)} + \mathbf{v}^{(n+1/2)}\Delta t$$

15   The Courant condition provides a limit for the maximum stable time step Δt, which can be written as

$$\Delta t < \frac{h}{c}$$

where h is the material sound speed, and c is the size of the smallest element in the model.

20   An advantage of the explicit method is that relatively small amount of computer CPU and memory resources are required, since a stiffness matrix is not used. Another

advantage of the explicit method is that convergence is not a problem, since no equilibrium iterations are necessary.

A disadvantage of the explicit method is that static equilibrium can only be approximated. Another disadvantage of the explicit method is that the Courant time

5   step size limit requires potentially hundreds of thousands of time steps to be computed during each simulation, which significantly slows the solution process. Yet another disadvantage of the explicit method is that techniques for accelerating the solution process, such as mass scaling, introduce artificial dynamic effects which are difficult to quantify and control.

10   As mentioned above, conventional simulations are performed using either implicit method finite element software or explicit method finite element software. However, simulations performed using implicit method finite element software suffer from the disadvantages of the implicit method mentioned above, and simulations performed using explicit method finite element software suffer from the disadvantages

15   of the explicit method mentioned above.

Thus, what is needed is a method that overcomes the disadvantages of conventional finite element simulations.

SUMMARY OF THE INVENTION

20   The present invention includes a method that overcomes the disadvantages of conventional finite element simulations. The method accomplishes this by switching between an implicit method and an explicit method one or more times during a finite element simulation.

In one embodiment of the present invention, a method for performing a finite element simulation is disclosed. The method includes switching between an implicit method and an explicit method one or more times during the finite element simulation.

In another embodiment of the present invention, a computer readable storage

5   medium storing one or more computer programs for performing a finite element simulation is disclosed. The computer programs include instructions for automatically switching between an implicit method and an explicit method one or more times during the finite element simulation.

In another embodiment of the present invention, a computer system is disclosed.

10   The computer system includes one or more computers and one or more computer programs running on the computers where the computer programs are for performing a finite element simulation. The computer programs include computer instructions for automatically switching between an implicit method and an explicit method one or more times during the finite element simulation.

15   In another embodiment of the present invention, a data signal embodied in a carrier wave is disclosed. The data signal includes one or more computer programs for performing a finite element simulation. The computer programs include instructions for automatically switching between an implicit method and an explicit method one or more times during the finite element simulation.

20   Other embodiments, aspects, and advantages of the present invention will become apparent from the following descriptions, the accompanying drawings, and the accompanying claims.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and for further features and advantages, reference is now made to the following description taken in conjunction with the accompanying drawings, in which:

5   FIG. 1 is a block diagram of an exemplary computer system upon which the methods of the present invention can be performed, according to an embodiment of the present invention.

FIG. 2 is a flowchart of an exemplary automatic implicit-explicit switching method, according to an embodiment of the present invention.

10   FIG. 3 is a flowchart of an exemplary automatic implicit-explicit switching method, according to an embodiment of the present invention.

FIGS. 4A and 4B show a switching procedure that was automatically generated using a method of the present invention to solve a simulation of the formation of a Budd Complex Channel.

15   FIG. 4C shows the Budd Complex Channel that results from the simulation of FIGS. 4A and 4B.

FIG. 4D shows a switching procedure that was discovered by trial-and-error using the manual switching method to solve a simulation of the formation of a Budd Complex Channel.

20   FIG. 5A shows a switching procedure that was automatically generated using a method of the present invention to solve a sheet metal springback simulation of the NUMISHEET '93 U-Channel part.

FIG. 5B shows the NUMISHEET '93 U-Channel part immediately after stamping (shaded portion) and after springback has occurred (unshaded portion).

25

DETAILED DESCRIPTION OF THE INVENTION

The preferred embodiments of the present invention and their advantages are best understood by referring to FIGS. 1 through 5 of the drawings. Like numerals are used for like and corresponding parts of the various drawings.

5

System Overview

FIG. 1 is a block diagram of an exemplary computer system 2 upon which the methods of the present invention can be performed, according to an embodiment of the present invention. Computer system 2 can include one or more suitable processing

10 facilities such as microcomputers/workstations, minicomputers, mainframes, and/or massively parallel processing computers running suitable operating systems. Examples of suitable microcomputers/workstations include the IBM PC from IBM Corporation of Armonk, N.Y., and the Sparc-Station from Sun Microsystems Corporation of Mountain View, Calif. Examples of suitable minicomputers are the VAX 750 from Digital

15 Equipment Corporation of Maynard, Mass. and the AS/400 from IBM. An example of a suitable mainframe is the System-390 series from IBM. Examples of suitable massively parallel processing computers include the Cray T3E or the Cray T90 from Cray Corporation of Mendota Heights, Minn. Examples of suitable operating systems include any of the various versions of Unix, VMS from Digital Equipment, or OS/2

20 from IBM. Computer systems used in the development of the invention include the IBM RS-6000, the Hewlett-Packard V-Class, and the Silicon Graphics Origin 2000, all of which are engineering workstations or servers running under the UNIX operating system, as ported by each hardware vendor. Those skilled in the art will recognize that other types of suitable computer systems and suitable operating systems can be used in

25 accordance with the present invention.

Computer system 2 includes the following components: a central processing

unit (CPU) 4, a random access memory (RAM) 6, a read-only memory (ROM) 8, a

storage device 10, an input/output (I/O) device 12, and a display 14. Although only a

single instance of each component is shown, it should be recognized that computer

5   system 2 can include one or more of any of the components shown in FIG. 1.

Computer system 2 can also include other suitable computer components. For clarity,

interconnections between the various components of computer system 2 are not shown.

CPU 4 can be any type of processor, such as a Pentium processor from Intel

Corporation or an Alpha EV5 from Digital Equipment Corporation, used to process

10   information in computer system 2.

RAM 6 can be any type of memory device, including dynamic and/or static

RAM, used to store any number of different types of information required by computer

system 2. For example, RAM 6 can store computer programs for performing the

method of the present invention.

15   ROM 8 can be any type of read-only memory device, including PROMS,

EPROMS and EEPROMs, used to store any number of different types of information

required by computer system 2. For example, ROM 8 can store information concerning

initialization routines or even some or all of the computer programs for performing the

method of the present invention.

20   Storage device 10 can be any type of mass storage device, including magnetic,

electrical, and optical-type storage devices, used to store any number of different types

of information required by computer system 2. For example, storage device 10 can

store various types of data or even some or all of the computer programs for performing

the method of the present invention.

I/O device 12 can be any type of input/output device, including keyboards,

printers, or any other type of device, by which information can be entered into or

received from computer system 2.

Display 14 can be any type of display device, including those of the CRT and

5   flat-panel display type, used to display information. For example, display 14 can

display the results of a computer simulation, which results from performing the

methods of the present invention, via a graphical user interface.

Implicit-Explicit Switching Overview

10   The implicit-explicit switching method of the present invention is a new

approach to finite element simulation of the mechanical response of structures. By

combining the existing implicit and explicit methods within a single simulation, and

switching back and forth between them, an improved solution can be obtained. The

solution is improved because the switching procedure avoids deficiencies in both the

15   implicit and explicit methods, producing a solution free of dynamic effects and

convergence difficulties.

The proposed implicit-explicit switching method is designed to avoid the

problems of both methods while capitalizing on their advantages. It allows the robust,

reliable explicit method to be used in support of the accurate, but less reliable implicit

20   method. In the implicit-explicit switching method, the implicit method is generally

employed initially, and continues to be used until convergence trouble is detected. At

this point, the solution is switched and the explicit method is used to advance the

solution past the troublesome spot. The solution then returns to the implicit method

and proceeds, switching repeatedly back and forth as necessary to avoid any further

25   convergence difficulties as they are encountered.

11

Convergence trouble is detected whenever the number of equilibrium iterations exceeds a threshold limit or when the internal energy of the system exceeds a threshold limit. With the implicit method alone, the simulation would be forced to terminate unsuccessfully at this point, or to back up in time and try to proceed with a smaller time

5   step size. Instead, the implicit-explicit switching method of the present invention activates a switch to the explicit method, and proceeds reliably since in the explicit method successful convergence of equilibrium iterations is not required.

The power of the implicit-explicit switching method comes from the ability of a brief explicit simulation interval to circumvent the convergence difficulty which often

10   halts implicit method simulations. Once the solution is switched back to the implicit method, static equilibrium iterations are resumed. Convergence is usually obtained easily, eliminating the inertial effects introduced during the explicit phase and re-establishing static equilibrium. In cases where convergence is not obtained, another switch to the explicit method is activated. The method continues switching until

15   convergence can be regained with the implicit method.

The implicit-explicit switching method is further described below in the sections entitled "Automatic Switching" and "Manual Switching."

Automatic Switching

20   FIG. 2 is a flowchart of an exemplary automatic implicit-explicit switching method 200, according to one embodiment of the present invention. Method 200 can be incorporated into a finite element analysis computer program, such as LS-DYNA produced by Livermore Software Technology Corporation, of Livermore, Calif. The computer program can be stored on a computer readable storage medium such as RAM

25   6, ROM 8, and/or storage device 10 of computer system 2, and performed on a

computer such as computer system 2. The computer program can also be transmitted between two or more computers over a network in a data signal embodied in a carrier wave.

Before method 200 begins, a user defines several parameters. First, a user

5   defines the implicit time step size $\Delta t^{implicit}$. Second, the user defines the explicit time step size $\Delta t^{explicit}$ which, as described above, is limited by the Courant condition. Third, the user defines the total simulation time t^{end}. Fourth, the user defines the iteration limit i^{max} and the maximum increase in internal energy E^{max} for the implicit method. These values are defined such that convergence problems can be detected (as discussed

10   below). Fifth, the user defines the maximum time the explicit method can operate $t^{explicit\_max}$. Sixth, the current time of the simulation t is set equal to the implicit time step size $\Delta t^{implicit}$.

In step 202, a simulation begins using the implicit method. At this time, an implicit method iteration counter is set to 1. In step 204, an implicit solution is

15   computed. As described above, this is accomplished by assembling a stiffness matrix, which represents the interaction of nodal motions within the subject structure, computing the inverse of the stiffness matrix, and then applying this inverse to an array of nodal forces to produce a solution of nodal displacements $\mathbf{u}^i(t)$. The solution of nodal displacements $\mathbf{u}^i(t)$ defines the position of each node in the system due to the

20   application of the nodal forces.

In step 206, the nodal displacements are tested to verify that they satisfy the governing equations of motion. If the equations of motion are satisfied (i.e., a satisfactory solution is obtained), the solution is said to have converged and method 200 advances to step 208. In step 208, the current time t is compared to the total

25   simulation time t^{end} defined by the user. If the current time t is equal to the total

simulation time t^{end}, then method 200 ends. If the current time t is less than the total

simulation time t^{end}, method 200 proceeds to step 210. In step 210, the current time t is

incremented by the implicit time step size $\Delta t^{implicit}$ and method 200 proceeds to step

202.

5   Referring back to step 206, if the equations are not satisfied (i.e., a satisfactory

solution is not obtained), method 200 proceeds to step 212. In step 212, the iteration

limit i^{max} and the maximum increase in internal energy E^{max} for the implicit method are

tested. A large number of iterations, or an excessive increase in internal energy,

indicates that the solution may not converge or that the solution may take an

10   unacceptable amount of time to converge. Both situations are undesirable.

The increase in internal energy is computed by comparing the internal energy of

the model at the beginning of the current time step with the internal energy of the

model at the end of the current iteration. The internal energy at the beginning of the

current time step is the product of the force at the beginning of the current time step

15   times the displacement at the beginning of the current time step. The internal energy at

the end of the current iteration is the product of the force at the end of the current

iteration times the displacement at the end of the current iteration. If the increase in

internal energy exceeds the maximum increase in internal energy E^{max} (which, for

example, may be 1,000,000 times the initial internal energy), this indicates that the

20   solution may not converge or that the solution may take an unacceptable amount of

time to converge.

If neither the iteration limit i^{max} nor the maximum internal energy E^{max} has been

exceeded, method 200 proceeds to step 214 where the current iteration i is incremented

and method 200 then proceeds to step 204. If either the iteration limit i^{max} or the

25   maximum energy limit E^{max} has been exceeded, method 200 proceeds to step 216.

14

In step 216, the simulation automatically switches from the implicit method to the explicit method. In this step, the current time t is reduced by the implicit time step size $\Delta t^{implicit}$ since the simulation during the previous implicit time step was unsuccessful.

5   In step 218, the explicit method is initialized. This is accomplished by initializing the position, velocity, and acceleration of each node in the system. The velocity and acceleration of each node in the system must be provided since the explicit method performs a dynamic (not static) simulation.

The position of each node in the system can be determined by using the position

10   of each node point from the last successful implicit step. The last successful implicit step is the last step performed using the implicit method where neither the iteration limit i^{max} nor the maximum energy limit E^{max} were exceeded. The velocity and acceleration of each node in the system can be determined by several different techniques. Two such techniques are described below, although it should be

15   recognized that other initialization techniques can be used in accordance with the present invention.

One technique for selecting the initial velocity and acceleration of each node point is to simply specify all values to be zero. This simple approach is intuitively appealing because the explicit dynamic method then begins with the model completely

20   at rest, free from all dynamic oscillations. However, it was discovered that this approach can induce undesirable dynamic oscillations later in the explicit phase of the simulation. These oscillations occur because any further motion of a node point creates non-zero velocity, and this change in velocity creates an accompanying acceleration, introducing dynamic inertia effects.

15

Another technique by which the initial velocity and acceleration of each node point can be determined is by computing the velocity of each node point from the last successful implicit step, and defining the acceleration of each node point as zero. The initial velocity of each node is computed as the change in displacement over the last

5   successful implicit step Δu^{last}, divided by the implicit time step size $\Delta t^{implicit}$ as follows:

$$v = \Delta u^{last}/\Delta t^{implicit}$$

If no prior successful implicit step exists, the initial velocity of each node is defined to be zero. Applicant has discovered that this technique reduces the dynamic error which is introduced into the explicit method, since further deformation of the

10   model along the previous direction is possible without changing the nodal velocity, hence without introducing any acceleration nor any accompanying inertial effects. Once the initial velocity and acceleration of each node point has been determined, method 200 proceeds to step 220.

In step 220, the time when the solution will automatically switch back to the

15   implicit method or the switch time t^{switch} is computed. This time is computed by adding to the current time t a maximum time the explicit method can operate $t^{explicit\_max}$. After computing the switch time t^{switch}, method 200 proceeds to step 222.

In step 222, the simulation proceeds using the explicit method. As described above, the explicit method operates by assembling forces at each node point,

20   determining the mass of each node point, and then dividing the forces at each node point by the mass of each node point to determine the acceleration of each node point. This information is then used to determine the solution of nodal displacements $u(t)$, (i.e., the position of each node in the system).

In step 224, after an explicit solution has been obtained, method 200 checks to

25   see if the end time t^{end} has been reached. If the current time t is equal to the end time

t^{end}, then method 200 ends. If the end time t^{end} has not been reached, method 200 checks to see if the switch time t^{switch} has been reached (step 226). If the switch time t^{switch} has not been reached, method 200 increments the current time t (step 228) by the explicit time step size $\Delta t^{explicit}$ and proceeds to step 222 to perform another explicit

5   solution. If the switch time t^{switch} has been reached, method 200 proceeds to step 230.

In step 230, the simulation automatically switches from the explicit method to the implicit method. In this step, the current time t is incremented by the implicit time step size $\Delta t^{implicit}$.

In step 232, the implicit method is initialized. This is accomplished by

10   initializing the position of each node in the system. The position of each node in the system is defined using the solution of nodal displacements $u(t)$ from the last explicit step performed.

Method 200 then proceeds to step 202 where current iteration i is set to 1, then to step 204 where an iteration using the implicit method is performed, then to step 206

15   where it is determined whether or not a satisfactory solution has been obtained, and then to step 212 where it is determined whether the iteration limit i^{max} or the maximum internal energy E^{max} has or has not been exceeded. If neither the iteration limit i^{max} nor the maximum internal energy E^{max} has been exceeded, method 200 proceeds to step 214 where the current iteration i is incremented and method 200 then proceeds to step 204.

20   If either the iteration limit i^{max} or the maximum energy limit E^{max} has been exceeded, method 200 proceeds to step 216 where the simulation automatically switches from the implicit method to the explicit method, and then to steps 218, 220, and 222, as described previously, where the simulation proceeds using the explicit method. This procedure continues until the simulation time is advanced to end time t^{end}, using a

25   combination of implicit and explicit methods as necessary according to method 200.

17

FIG. 3 is a flowchart of an exemplary automatic implicit-explicit switching

method 300, according to one embodiment of the present invention. Like method 200,

method 300 can be incorporated into a finite element analysis computer program, such

as LS-DYNA produced by Livermore Software Technology Corporation, of Livermore,

5   Calif. The computer program can be stored on a computer readable storage medium

such as RAM 6, ROM 8, and/or storage device 10 of computer system 2, and

performed on a computer such as computer system 2. The computer program can also

be transmitted between two or more computers over a network in a data signal

embodied in a carrier wave.

10   Method 300 is a modification of method 200. Steps 302-320 and 326-332 of

method 300 are performed in a manner similar to steps 202-220 and 226-232 of method

200. Method 300 is different from method 200 in that method 300 forces the final step

of the simulation to be performed using the implicit method, whereas, according to

method 200, the final step in the simulation can be performed using the implicit method

15   (step 208) or the explicit method (step 224). Method 300 is useful in simulations that,

to be accurate, require a static solution. For example, method 300 is useful for

springback simulations.

Method 300 introduces a new step after step 320, where the switch time t^{switch}

has been computed. In the new step 321, the end time t^{end} of the simulation is

20   automatically extended beyond the switch time t^{switch} by the duration of one implicit

time step $\Delta t^{implicit}$. The end time t^{end} of the simulation is extended to ensure that the

switch time t^{switch} will always be reached before the end time t^{end} when the explicit

method is in use, and allows the end time check to be removed from the explicit

solution loop (see step 224 of method 200). In this way, the end time t^{end} can only be

reached from the implicit loop via steps 306 and 308, and thus the final step is guaranteed to be reached using the implicit method.

Manual Switching

5   Although methods 200 and 300 have described automatic implicit-explicit switching methods, the scope of the present invention also encompasses manual implicit-explicit switching methods. To perform a manual switching simulation, two parameters must be determined: the exact times to switch between the implicit and explicit methods and the number of times to switch between the methods. These

10   parameters are obtained through a trial-and-error process. Generally, numerous erroneous simulations must be performed before acceptable parameters are found for a particular simulation and thus only simplistic switching schemes can be generated in a practical effort (e.g., only switching two or three times during a simulation). As a result, the manual switching method is typically only useful for simplistic simulations.

15

Exemplary Applications

FIG. 4A shows a switching procedure that was automatically generated using method 200 of the present invention, as applied to a simulation of the formation of the Budd Complex Channel. The Budd Complex Channel is a well known benchmark part

20   used in computer simulations. The solution method that is in use (i.e., either the implicit or explicit method) and the time step size used in each method is shown on the y-axis, and the elapsed time of the simulation t is shown on the x-axis. During the implicit phases of the simulation, the time step size is large (time step size = 0.001 seconds) whereas during the explicit phases the Courant condition limits the time step

25   size to a very small value (1.8×10^{-6} seconds).

The simulation begins at time t = 0 using the implicit method. At time t = 0.007 seconds the implicit method encounters convergence trouble. After the number of iterations using the implicit method reaches the iteration limit i^{max} or the maximum internal energy E^{max}, the simulation automatically switches to the explicit method. The

5   simulation then proceeds for a brief, predetermined period of time (i.e.,the explicit interval $t^{explicit\_max}$) using the explicit method, which in this example was provided as an input data value of 0.0005 seconds. Then the simulation automatically switches back to the implicit method. This process of automatically switching back and forth between the implicit and explicit methods continues until the simulation proceeds to the end

10   time of 0.078 seconds. The size of the last implicit step is automatically adjusted to reach the end time exactly.

FIG. 4B shows a detailed view of the switching activity which occurs early in the simulation. From FIG. 4B, it can be seen that the first implicit convergence trouble was detected at a simulation time of 0.0070 seconds. Method 200 automatically

15   switched the simulation to the explicit method, and proceeded with this method until a simulation time of 0.0075 seconds. At this time, method 200 automatically switched the simulation back to the implicit method, however convergence trouble was again encountered. Method 200 automatically switched the simulation back to the explicit method, and proceeded to a simulation time of 0.0080 seconds. At this time, method

20   200 automatically switched the simulation again to the implicit method, which proceeded without convergence difficulty until simulation time 0.0090 seconds. Method 200 activated an automatic switch to the explicit method, and automatically returned to the implicit method at simulation time 0.0095 seconds, where the implicit method was again able to proceed without convergence trouble. Further convergence

25   trouble was automatically detected at simulation times of 0.0155 seconds and 0.0200

seconds. Method 200 automatically applied switches to the explicit method once and thrice, respectively, to overcome these convergence difficulties. FIG. 4C shows the Budd Complex Channel that results from the simulation.

5 FIG. 4D shows a switching procedure that was manually generated using the present invention, as applied to a simulation of the formation of the Budd Complex Channel. In particular, FIG. 4D shows the manual switching scheme during the early stages of the Budd Channel forming simulation. This relatively simple switching scheme required over ten trial-and-error simulations to develop, as the simulation times for each of the three switches had to be discovered. Notice that the explicit method was

10 used during the initial phase of the simulation, since a successful strategy which began with the implicit method could not be discovered by an experienced operator through a reasonable number of trial-and-error iterations.

Although both the manual and the automatic switching methods provide advantages over conventional implicit only or explicit only simulation techniques, the

15 automatic switching method provides several benefits over the manual switching procedure. For example, the automatic switching method produces a solution much faster than that which could be obtained using the manual switching procedure. Using the automatic switching method, the Budd Complex Channel simulation was completed in approximately 31 CPU hours on an IBM RS-6000/260 workstation computer. Using

20 the manual switching procedure, a similar simulation was completed in approximately 27 CPU hours on the same computer. However, the trial-and-error process of determining the switching scheme for the manual method involved more than ten failed simulations. Thus the total simulation time using the manual switching method, including the trial-and-error process, was approximately ten times larger than the total

25 simulation time using the automatic switching method. The cost savings measured in

21

terms of man-hours required to conduct the simulation was even more significant, since

the trial-and-error process required by the manual switching method was extremely

labor intensive, and required an experienced operator, while the switching procedure

generated by the automatic switching method ran successfully on the first attempt

5   without intervention by the operator.

Additionally, the automatic switching method produces a solution that is far

more complex than that which could realistically be obtained using trial-and-error with

the manual switching procedure. For example, the simulation of the Budd Complex

channel using the automatic switching method involved fourteen switches between the

10   implicit method and explicit method, while the manual switching scheme developed in

over ten trial-and-error simulations included only three switches. FIG. 4D shows the

switching scheme developed in the manual method, as compared to FIG. 4B for the

automatic method. A complex solution using fourteen switches would have been

virtually impossible to discover by trial-and-error using the manual switching

15   procedure.

FIG. 5A shows a switching procedure that was automatically created using

method 300 to solve a sheet metal springback simulation of the NUMISHEET '93 U-

Channel part. Springback is a problem that occurs when a stamped metal part is

removed from formation tooling, and the part changes shape or deforms as internal

20   stresses are relieved. Deformation due to springback creates a final part which no

longer matches the shape of the tooling with which it was produced and, thus, is

undesirable. Sheet metal parts which exhibit a large amount of deformation due to

springback are typically very flexible. These types of parts are particularly challenging

for the implicit method, and convergence of the equilibrium iteration process is

22

inherently difficult. Thus, the implicit-explicit switch is ideally suited for springback simulation.

Referring to FIG. 5A, the y-axis (time step size) indicates which simulation method is employed: The implicit method uses a large time step (1.0 to 2.0

5   milliseconds), whereas the explicit method uses a small time step (9.0×10^{-4} milliseconds). During the simulation, the implicit method ran successfully during the first four time steps, but encountered convergence difficulty on the fifth time step. The fifth time step was an attempt advanced the solution to the original simulation end time of 10.0 milliseconds. The convergence failure was detected by method 300, and the

10   explicit method was automatically activated for a duration of 1.0 milliseconds. Then the method 200 automatically switched back to the implicit method, and another implicit step was attempted. This step failed to converge also, so method 300 switched back to the explicit method and advanced the solution for a duration of 1.0 ms.

At this point the solution reached the original simulation end time, but since the

15   current solution method was not implicit, method 300 automatically extended the end time by 1.0 milliseconds before switching back to the implicit method. This extension of the solution end time ensures that the last step of the simulation will be performed with the implicit method, and represents the fundamental difference between method 200 and method 300.

20   Several more automatic switches to the explicit method were performed, each for a duration of approximately 1.0 milliseconds, and each time returning to test for convergence with the implicit method. After a total of 31 switches, the part deformation had proceeded dynamically to a state which was close enough to the static equilibrium solution that the implicit method could finally converge, and the solution

25   finished successfully.

23

The shaded portion of FIG. 5B shows the NUMISHEET '93 U-Channel part immediately after stamping or forming. The part geometry immediately after stamping includes four 90-degree angles $\theta 1$, $\theta 2$, $\theta 3$, and $\theta 4$, a horizontal center section 502, vertical sidewalls 503 and 504, and horizontal flanges 505 and 506. The unshaded

5   portion of FIG. 5C shows the NUMISHEET '93 U-Channel part of FIG. 5B after springback has occurred (as simulated). After springback, $\theta 1$ is now less than 90 degrees, $\theta 2$ and $\theta 3$ are greater than 90 degrees, and $\theta 4$ is less than 90 degrees.

Advantages Of The Present Invention

10   Performing finite element simulations using the implicit-explicit switching methods of the present invention provides several advantages over finite element simulation techniques. First, the manual and automatic implicit-explicit switching methods provide a way to capitalize on the strengths of the implicit method and the explicit method while minimizing the weaknesses of each method. Second, the

15   automatic implicit-explicit switching method is a robust, automatic procedure which can obtain solutions which are very accurate and can be obtained with minimal operator interaction. Conventional simulation techniques often could not produce an accurate solution and required significant operator interaction.

While particular embodiments of the present invention and their advantages

20   have been shown and described, it should be understood that various changes, substitutions, and alterations can be made therein without departing from the spirit and scope of the invention as defined by the appended claims. For example, although it has been described how the present invention can be used in formation simulations and springback simulations, those skilled in the art will recognize that the present invention

25   can be applied to a wide variety of problems relating to non-linear static or dynamic

24

analysis using the finite element method, such as general structural mechanics and

stress analysis, fluid mechanics, biomechanics, heat transfer, geomechanics ,

aeromechanics, chemical reactions, acoustics, and electric and magnetic fields. As

another example, methods 200 and 300 can begin using the explicit method rather than

5   the implicit method.

10

15

20

25